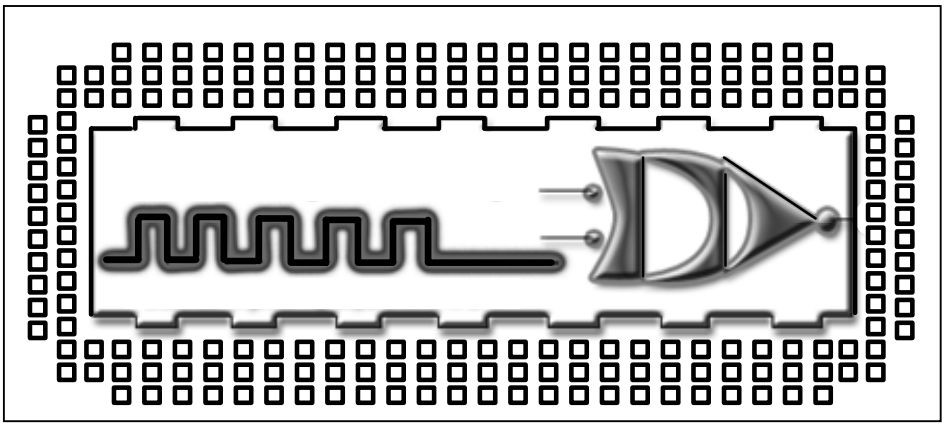
Department of Electrical Engineering
King Fahd University of Petroleum and Minerals
Dhahran - Saudi Arabia

Laboratory Manual
EE 200 Digital Design



August 2003

PREFACE

This document has been prepared to serve as a laboratory manual for EE 200 Digital Design course for electrical engineering students. The manual consists of a set of experiments designed to allow students to build, and verify digital circuits and systems. This set of experiments cover relevant topics prescribed in the syllabus and are designed to reinforce the theoretical concepts taught in the classroom with practical experience in the lab. By the end of the course, students are expected to have a good understanding of digital logic design and implementation with SSI and MSI devices.

LIST OF EXPERIMENTS

LAB GUIDELINES

PRE-LAB

Each student will do his own pre-lab. It is intended in this course to increase the student's utilization of PC; therefore, the pre-lab write up must be typed. This same write up should be modified to be submitted as the Lab report.

Starting from Lab 3, all pre-lab must be done using LogicWorks simulation package. During the Lab, each student may be asked about the simulation results. All circuit parts with pin numbers should be included in the pre-lab so that you will be ready to start connecting the hardware in the lab. Try to investigate all possible changes on the circuit to acquire full knowledge about your design. All questions in the Lab handout should be answered based on the LogicWorks results. The pre-lab will make 35% of the total grade of the Lab experiment.

THE LAB

During the lab, the students should work as a group. The Pre-Lab results from each student in the group will compared and the circuit the produces the best results will be implemented using hardware parts. Elaborate on your conclusion about the observations about the simulated and obtained results. Punctuality of attendance to the Lab is worth 5% and the active participation on the experiment will count as 30% of the total grade.

THE LAB REPORT

Each student should have his own report. The lab reports are intended to serve two equally important purposes. First, they indicate your technical comprehension of the topics addressed in the labs, and second, they indicate your ability to present and discuss your results in a clear and concise manner. You will be graded on both aspects of your report.

The suggested format for your lab report is given below.

1. **Objectives:** State clearly what you set out to achieve in this lab. If this differs from what you finally achieved, explain it in the "Conclusions" section. Please do not copy the objectives verbatim from the lab handout. Think about it, interpret it, and explain it the best you can, in your own words.

2. **Parts:** List all the parts you used in the design.

3. **Design and Test Procedure:** For *each subsection* of the lab, explain the following:

(a) Step-by-step description of what you did. Include as many details as possible, and once again, write it in your own words.

(b) All necessary calculations as well as all pin-to-pin circuit diagrams of your design. Please make sure your figures are consistent, legible and well labeled.

(c) Your testing procedure. Explain how you went about testing your design. Did you try testing critical individual blocks first?

4. **Results and Answers to Questions:** For *each subsection* of the lab, present your results in a clear and concise manner (label graph axes, include all units of measurement). Note down all your observations, even if you were not specifically asked for them in the handout. Interpret your results and discuss the accuracy of your measurements. Additionally, answer all questions listed in the lab handout.

5. **Conclusions**: In this section you should attempt to answer the questions: What did you learn from this lab? What did you do wrong (or what went wrong)? How could you have improved upon your design and test procedures? Were your results as expected or did you find something unusual. Try not to include information that you have included in previous sections. Present the significance of your results conceptually, if applicable, (e.g. The CAD tool does not capture the glitching behavior accurately.)

The Lab Report will count as 30% of the grade and is due at the beginning of the subsequent lab experiment.

EE 200 DIGITAL LOGIC DESIGN EXPERIMENT #0

INTRODUCTION TO LAB EQUIPMENT

OBJECTIVE:

- To get acquainted with the breadboard and the cathode ray oscilloscope.

APPARATUS:

- Dual –trace oscilloscope
- Digital Proto-Board

THEORY:

See sections 1-2, and 1-6 in the book.

PB-503-C Analog/Digital Proto-Board:

The PB-503-C Analog/Digital Proto-Board is a self-contained digital logic laboratory. It includes a +5 volt power supply that provides operation power to the circuits under test, and also serves a ‘’1’’ logic level for TTL (transistor-transistor logic) integrated circuits. The ‘’0’’ logic level is represented by connection ground. Located on the front panel (see Fig 1) is a Breadboarding Socket that contains over 2500 tie points. These tie points are divided into 384 sets of five electrically interconnected solderless tie points, 8 sets of 25 interconnected solderless tie points along the right and left edges, and 4 sets of 50 interconnected solderless tie points on the top of the board. Tie points are spaced 0.1 inch apart and will accommodate the pins of DIP {dual-in-line package) integrated circuits, as well as a wide variety of other circuit components. The four groups of tie points (50 tie points each) at the top of the breadboarding are connected to +5V, an adjustable (+5, 15V), an adjustable (-5, -15V), and a ground connection, respectively. In the EE200 Lab experiments, we will only use the +5V row and the ground row of tie points. The eight larger groups of tie points (25 tie points each) are handy where large number of connections are to be made to a common circuit point, e.g., extending the ground, +5volt, etc.

Other useful features of the PB-503 include:

- Function Generator: The multi-waveform function generator provides continuously variable frequency signals from 0.1Hz to 100KHz. The frequency is selected in three ranges, with each range covering two-decades. The generator produces, sine, triangle, and square waveforms.

- Logic Indicators: A bank of eight LEDs is provided for use as built-in logic indicators. The LEDs are active high (they light) to indicate a “logic one” condition.

- Debounced Pushbuttons (Pulsers): two manual, bounceless (digitally conditioned) pulser buttons PB1 and PB2.

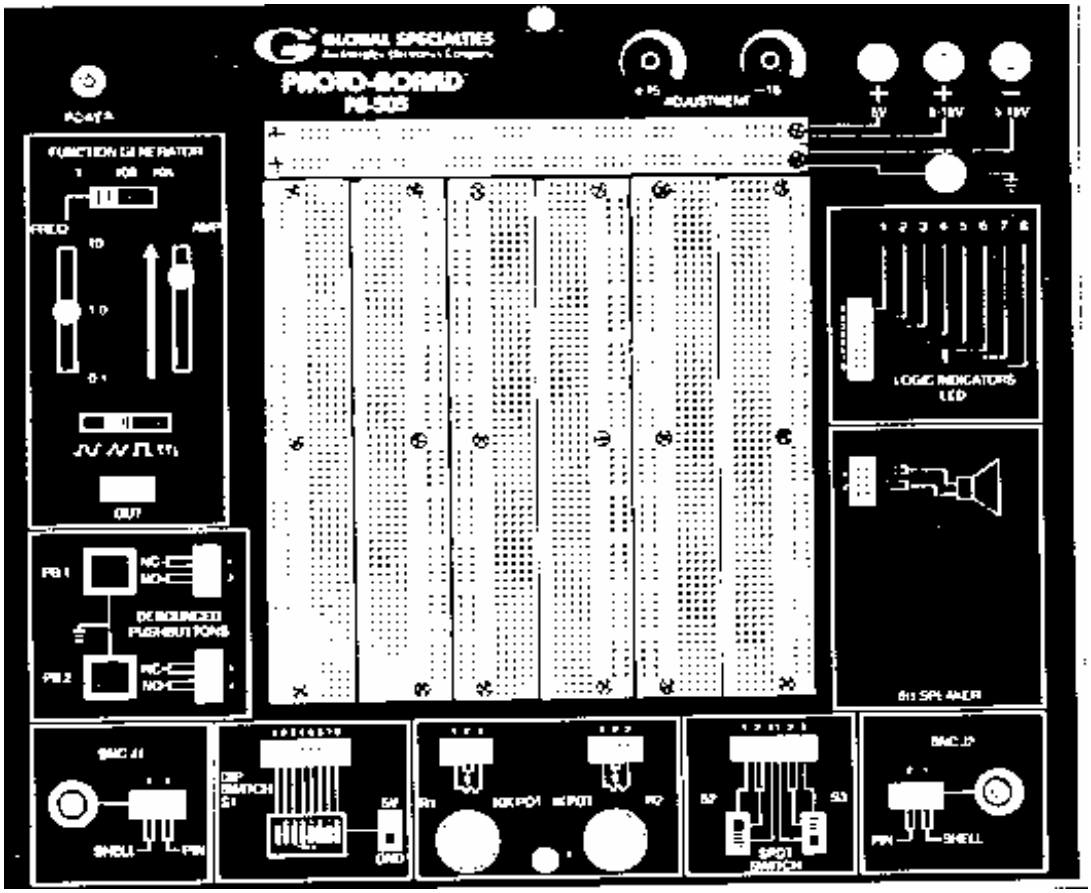
- Switches: An eight-pole DIP switch provides a convenient source of digital outputs. All eight switches have one side connected to a common lead, which may be switched to either +5 volts or ground. The remaining sides of all eight switches are separate, available, and uncommitted. This arrangement makes connecting special digital circuitry such as an eight-bit input port quick and easy. In addition to

the eight-pole switch, there are two single pole, double throw (SPDT) switches provided for general switching functions.

- Potentiometers: Two potentiometers are provided on the PB-503. The resistance values chosen (1 K and 10 K ohms) may be used in common circuit applications.

- BNC Connectors: The PB-503 may be connected to other pieces of equipment via two BNC connectors BNC J1 and BNC J2. These allow the use of shielded cable to minimize noise and interference.

Fig.1 PB-503 Panel layout.



IC PIN CONNECTIONs:

The IC type 7493 is in a 14-pin dual in-line case. The base pins progress in a counter-clockwise direction as seen from the side away from the pins, as shown Fig 2. Pin 1 is located by an identifying symbol, or the location of pins 1 and 14 are identified by an index notch at the end of the case where pins 1 and 14 are located.

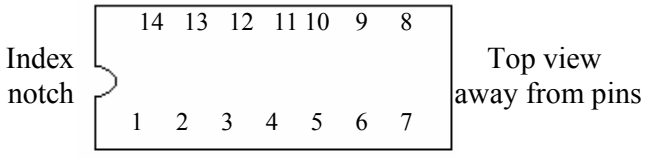


Fig.2 IC pin location, 14 pin dual-in-line (TO-116) case

PROCEDURE:

THE PB-503

1. Connect the PB-503 line cord into the AC power supply and turn on the power switch.

2. Connect the LOGIC INDICATORS (lamp monitor) (1, 2, …,8) to +5 volts. The lamps monitors should light when connected +5 volts and this "ON" lamp condition will represent a "l" logic level in your experiments.

3. Now connect the lamp monitors to ground. They should all be off at this time. This "OFF": lamp condition will represent a "0" logic level in your experiments.

4. Connect one side of a resistor (20 ohms, to 100 K) to ground and the other side to DIP switch S1-1 and switch the 5 V/GND switch to 5 volts position (These steps are already done for you). Connect S1-1 to LED-1. Now, when the S1-1 is pushed up to the closed position LED-1 will light, and when the S1-1 is brought back to the open position the LED will be off. Repeat these steps for S1-2 through S1-8 and observe the resultant condition of the lamp monitors.

Switch position	lamp	logic level
CLOSED	ON	1
OPEN	OFF	0

The switches can, thus, be used to supply logic level inputs to experimental circuits.

5. PULSER BUTTONS. Connect one side of a resistor (20 ohms, to 100 K) to +5 volts and the other side to PB1-1, the NC point. Then connect PB1-1 (the other lead of NC point) to LED-1. The LED should light when PB1 is pressed and extinguish when PB1 is released. Next, move the connections from PB1-1 to PB1-2, the NO point. Now the LED should be lit when PB1 is not pressed and go off when PB1 is pressed. Repeat these steps for PB2. These buttons will be used to enter momentary pulses of "O" and "1" logic levels.
6. Single Pole, Double Throw (SPDT) switches. Connect one side of a resistor (20 ohms, to 100 K) to +5 volts and the other side to lead 1 of S2. Then connect lead 2 to LED-1 and Lead 3 to LED-2. When the switch is brought to the up position then LED-1 and LED-2 will be ON and when the switch is brought down, the two LED's will be off. Repeat these steps on S3. These switches are provided for general switching functions.

7. CLOCK output. Connect the FUNCTION GENERATOR output TTL to LED-1. Set the function generator to "times 1" position and move the frequency control all the way to the top. Set the frequency selector to Hz. LED-1 should flash on and off, alternately at about 1 cycle per second. Move the function generator to "times 10" position (setting the frequency to 10 Hz). The lamp monitors should flash on and off at a faster rate, too high to count. Higher frequency settings "times 100" should cause the lamps to appear to be on continuously, at about half-normal brilliance

8. Connect the FUNCTION GENERATOR output TTL to an oscilloscope. You should observe a square wave having fairly steep sides and a peak-to-peak

amplitude of 4 to 5 volts. Change the selection to Square, triangle, Sine and observe the waves on the oscilloscope.

EE 200 DIGITAL LOGIC DESIGN EXPERIMENT #1

BINARY AND DECIMAL NUMBERS

OBJECTIVE:

- To demonstrate the count sequence of binary number and the binary-coded decimal (BCD) representation.

APPARATUS:

- IC type 7493 4-bit ripple counter

BINARY COUNT

1. Turn off the power switch.
2. Connect the IC type 7493 as shown in Fig. 3 Pin 14 is connected to PB1.
3. Turn the power on and observe the four indicator lamps. The 4-bit number in the out is incremented by one for every pulse generated by pushing the pulser button PB1
4. Disconnect the input of the counter at pin 14 from PB1 and connect it to the FUNCTION GENERATOR (lead TTL).
5. Set frequency selector to “time 1” (1 Hz). This will provide an automatic binary count.

THE BCD COUNT

1. Turn off the power switch.
2. Connect the IC type 7493 as shown in Fig.4 Pin 14 is connected to PB1.
3. Turn the power on and observe the four indicator lamps. The 4-bit number in the lambs is incremented by one for every pulse generated by pushing the pulser button PB1 following the sequence 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, ….
4. Disconnect the input of the counter at pin 14 from PB1 and connect it to TTL. Set frequency selector to “time 1” (1 Hz). This will provide an automatic binary count.

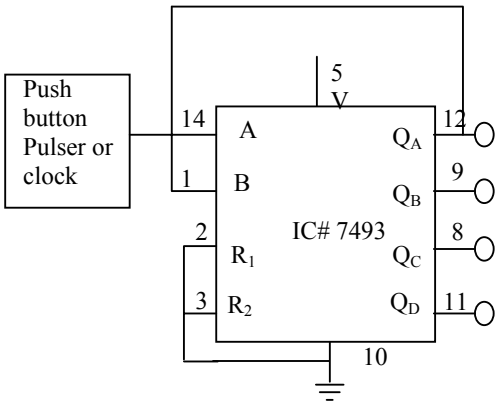


Fig.3 Binary Counter

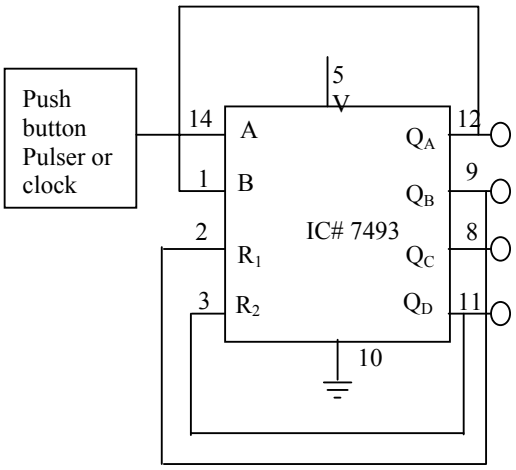


Fig.4 BCD counter

DIGITAL LOGIC GATES

OBJECTIVE:

- To study the basic logic gates: AND, OR, INVERT, NAND, and NOR.
- To study the representation of these functions by truth tables, logic diagrams and Boolean algebra.
- To observe the pulse response of logic gates.
- To measure the propagation delay of logic gates.

APPARATUS:

- IC Type 7400 Quadruple 2-input NAND gates
- IC Type 7402 Quadruple 2-input NOR gates
- IC Type 7404 Hex Inverters
- IC Type 7408 Quadruple 2-input AND gates
- IC Type 7432 Quadruple 2-input OR gates
- IC Type 7486 Quadruple 2-input XOR gate
- IC Type 7493 4-bit ripple counter
- Digi-Designer Logic Board
- Dual-trace oscilloscope

THEORY:

AND A multi-input circuit in which the output is 1 only if all inputs are 1.The symbolic representation of the AND gate is shown in Fig. 1a.

OR A multi-input circuit in which the output is 1 when any input is 1. The symbolic representation of the OR gate is shown in Fig. 1b.

INVERT The output is 0 when the input is 1, and the output is 1 when the input is 0. The symbolic representation of an inverter is shown in Fig. 1c.

NAND AND followed by INVERT. The symbolic representation of the NAND gate is shown in Fig 1d.

NOR OR followed by INVERT as shown in Fig 1e.

EX-OR The output of the Exclusive –OR gate, is 0 when it's two inputs are the same and it's output is 1 when its two inputs are different.

Truth Table Representation of the output logic levels of a logic circuit for every possible combination of levels of the inputs. This is best done by means of a systematic tabulation.

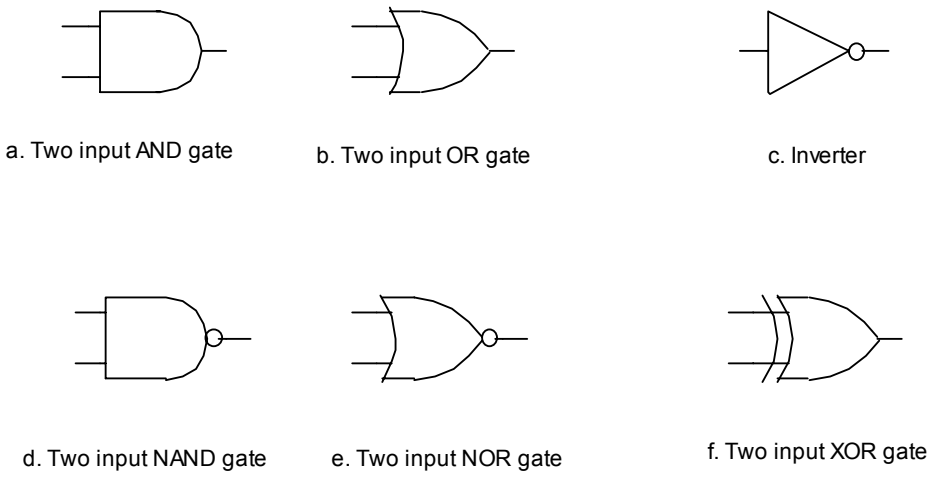


Fig.1 Symbols for digital logic gates

Part 1: Logic Functions

I. AND, OR, NAND, and NOR gates.

1. Use one gate for each IC 7400 (NAND), 7402 (NOR), 7408 (AND), 7432 (OR), 7486 (XOR). Each has input pins, 1 and 2, and output pin 3.

2. Connect pin 1 to switch S1-1, pin 2 to switch S1-2, and pin 3 to LED-1 for every gate as shown in Fig 2 as an example for the NAND gate.

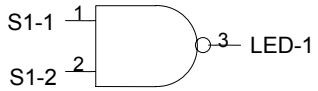


Fig.2 Two input NAND gate

3. Using logic switches S1-1 and S-2, apply the logic levels 0 and 1 to gate inputs (pin 1, pin 2), in the sequence shown in table 1. Record the output logic levels (see lamp LED-1) in table 1. Repeat the recordings for each gate.

Remember: Lamp ON = Logic 1, (High)
Lamp OFF = Logic 0 (Low)

Table 1

Pin 1	Pin 2	Pin 3

4. Use an inverter gate from IC 7404 whose input pin is pin 1 and whose output pin is pin 2.

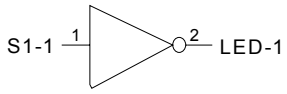


Fig.3 Inverter gate

5. Using logic switches S1-1, apply the logic levels 0 and 1 in the sequence shown in table 2. Record the output logic levels in table 2

Table 2.

Pin 1	Pin 2
0	
1	

Part-2: Response of Logic Gates:

Connect the circuits of figures 4 and 5 and write the corresponding truth tables 3 and 4, respectively.

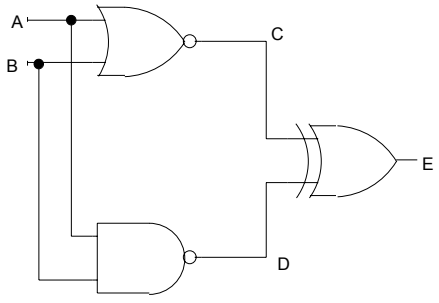


Fig. 4

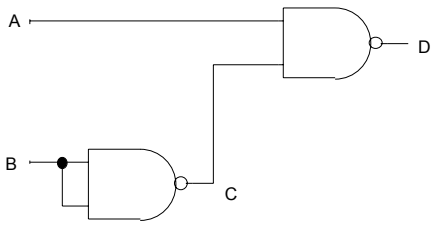


Fig. 5

Table 3.

A	B	C	D	E
0	0			
0	1			
1	0			
1	1			

Table 4.

A	B	C	D
0	0		
0	1		
1	0		
1	1		

Part-3: Propagation Delay in Logic Gates:

Connect all inverters inside two 7404 Ics in cascade. The output will be the same as the input except that it will be delayed by the time it takes the signal to propagate through all six inverters. Set S2 to 100 kHz and apply clock pulses to the input of the first inverter (connect pin 1 to j14) record the wave forms and determine the time delay from the input to the sixth inverter. This is done with a dual trace oscilloscope by applying the input clock pulses to one of the channels and the output of the sixth inverter to the second channel and measuring the delay between the two signals as shown in Fig 6. By using measured delay between two signals calculate the propagation delay for each inverter gate.

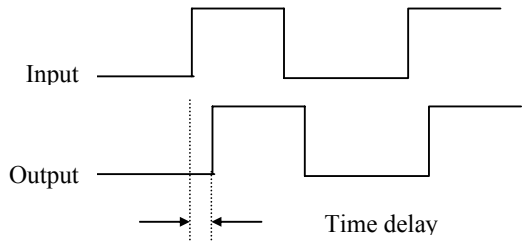
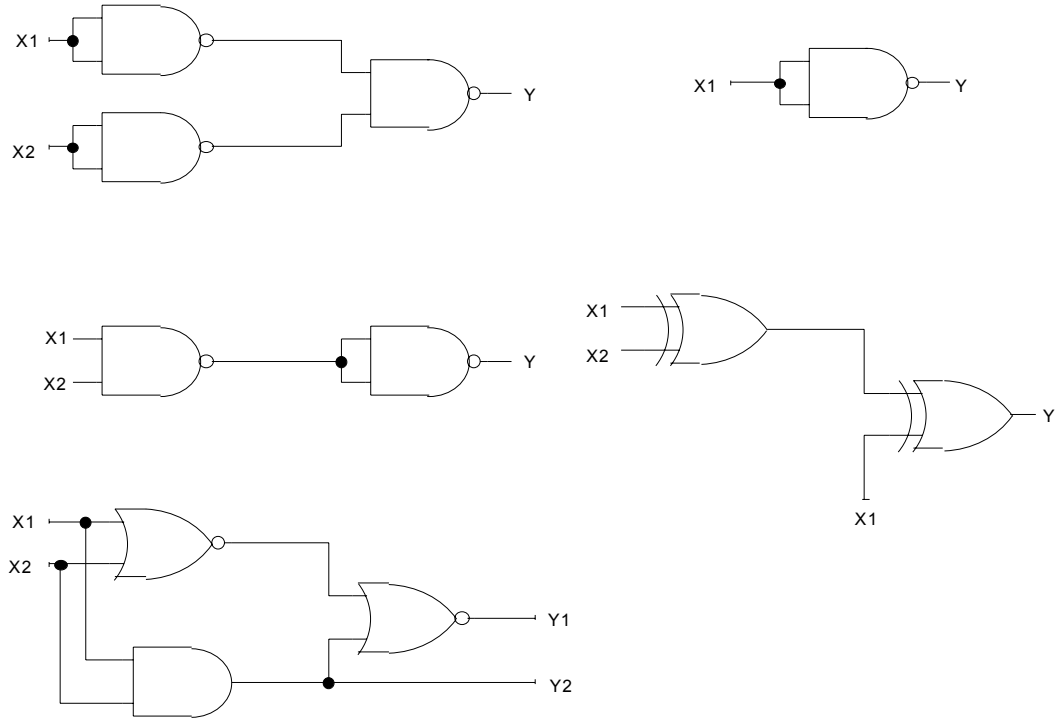


Fig. 6 Propagation delay

Part 4: Review Questions:

1. Write a truth table for each circuit. Derive Boolean expressions for all outputs.



2. A burglar alarm for a car has a normally low switch on each of four doors. If any door is opened the output of that switch goes HIGH. The alarm is set off with an active-LOW output signal. What type of gate will provide this logic? Support your answer with an explanation.

INTRODUCTION TO LOGICWORKS

(Handouts will be given to students)

BOOLEAN ALGEBRA

OBJECTIVE:

- To verify the rules and regulations of Boolean Algebra
- To simplify and modify Boolean logic functions by means of Demorgan's theorem.
- To design and implement a logic circuit.

APPARATUS:

- PB-503
- 7400 Quadruple 2 input NAND gates.
- 7402 Quadruple 2 input NOR gates
- 7408 Quadruple 2 input AND gates
- 7432 Quadruple 2 input OR gates
- 7404 Hex inverters
- 7411 Triple 3-input AND gate

THEORY: (See chapter 2 of the textbook)

1. A+0 = A
2. A+1 = 1
3. A .0 = 0
4. A .1 = A
5. A+A = A
6. A+A' = 1
7. A.A = A
8. A.A' = 0
9. (A')' = A
10. A+AB = A
11. A+A'B = A+B
12. (A+B)(A+C) = A+BC
13. A'. B' = (A+B)'
14. A'+B' = (A.B)'

Procedure 1:

a. Prove rule 1 using LogicWorks. The procedure is:
   - I. Open a new design window
   - II. Choose "ALL LIBRARY" in the Parts Palette
   - III. Put "OR" in the Filter window
   - IV. Select and double click on OR-2
   - V. Move to the cursor back into the circuit window. The cursor on the screen will now be replaced by a moving image of an OR gate.

VI. Position the OR gate near the center of the circuit window and click the mouse button.
VII. Press the spacebar to return to point mode.
VIII. Move again to the Parts Palette and type on the Filter "switch" or part of the word switch e.g. "sw".
IX. Select Binary switch and connect it to an input of the OR gate in the design window. (If you want to move the binary switch around, press the shift key while moving it).
X. Move again to the Parts Palette and select ground to be connected to the other input of the OR gate.
XI. Using the same method get a Binary Probe and connect it to the output of the OR gate
XII. Click on the binary switch to change it between 0 and 1 and notice how the rule A+0 = A is satisfied.

In the lab connect the circuit as shown in the figure using the switch S1-1 and LED-1 to verify the rule.

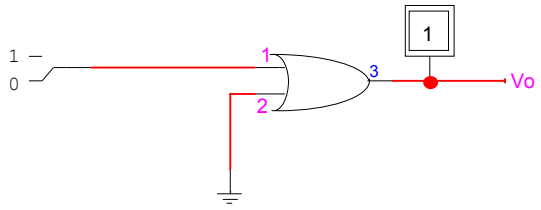


Fig.1 Verifying Rule 1

b. Connect the circuit of Fig.2 Using LogicWorks. Which rule does this circuit illustrate?

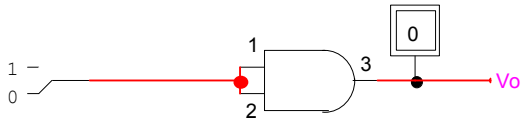


Fig.2

In the lab connect the circuit as shown in the figure using the switch S1-1 and LED-1 to verify the rule.

c. Design a circuit that illustrates rule 10. Use clock generator of the PB-503 for A and one of the logic switches of S1 for B. Copy the circuit from LogicWorks and paste it in your lab report.
d. Rule 6 illustrates that A+A' could be replaced with a wire to Vcc. What does rule 8 illustrate?
e. Rule 11 states that A+A'B = A+B. Using LogicWorks design a circuit that illustrates each of these expressions.

A+A'B
A+B

Prove that these two circuits perform equivalent logic. (Connect two circuits and show that their outputs are the same).

Procedure 2: Demorgan's Theorem

Proof of equation (1)

Using LogicWorks construct the two circuits given in Figs.3 and 4 corresponding to the functions A'. B'and (A+B)' respectively.

Show that for all combinations of A and B, the two circuits give identical results.

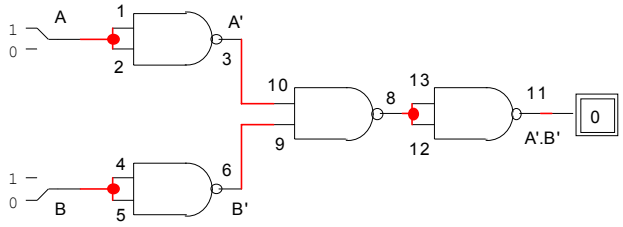


Fig.3

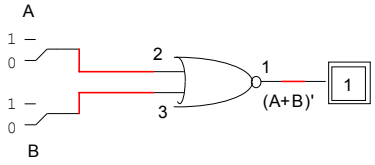


Fig.4

Proof of equation (2)

Using LogicWorks construct two circuits given in Figs. 5 and 6, corresponding to the functions A'+B' and (A.B)' A.B, respectively.

Show that, for all combinations of A and B, the two circuits give identical results.

In the lab connect these circuits and verify their operations.

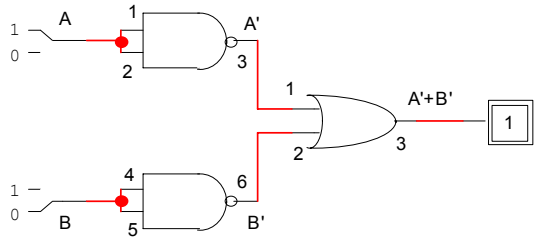


Fig. 5

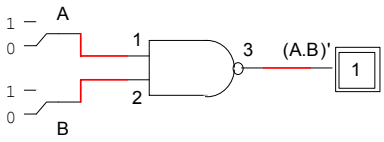


Fig. 6

II. Design of a Digital Circuit

Consider the following problem:

Four chairs A, B, C, and D are placed in a row. Each chair may be occupied ("l") or empty ("0"). A Boolean function F is "l" if and only if there are two or more adjacent chairs that are empty.

1. Give the truth table defining the Boolean function F
2. Express F as a minterm expansion (standard sum of product)
3. Express F as a maxterm expansion (standard product of sum

4. Using postulates and theorems of Boolean algebra, simplify the minterm expansion of F to a form with as few occurrences of each as possible.

5. Implement on LogicWorks for the pre-lab and then on PB-503, the simplified Boolean function with logic gates and check the operation of the circuit.

Notes:

- In LogicWorks use Binary Switches to represent the four chairs and connect the output of the circuit to a Binary Probe. Check that the Probe is "1" if and only if there are two or more adjacent chairs that are empty.

- For the hardware circuit in the lab, use logic switches S1-1, S1-2, S1-3, and S1-4 to represent the chairs and connect the output of the circuit to LED-1

Result:

Show all truth tables, circuits (using LogicWorks), etc. used in completing this experiment.

SIMPLIFICATION OF BOOLEAN FUNCTIONS USING K-MAP TECHNIQUES

OBJECTIVE:

- **To develop the truth table for a combinational logic problem**
- To use Karnaugh map to simplify Boolean expressions.
- To draw and simplify sum of products expressions.
- To draw logic diagrams using NAND gates.

APPARATUS:

- PB-503
- 7400 Quadruple 2 input NAND gates.
- 7404 Hex inverters
- 7410 Triple 3-input NAND gates
- 7420 Dual 4-input NAND gates

THEORY:

See chapter 3 of the text," simplification of Boolean functions"

Procedure:

Part 1: BCD invalid code detector

BCD is a 4-bit binary code representing the decimal numbers 0 through 9.

The binary numbers 1010 through 1111 are not used in BCD.

a) Construct a truth table containing all possible inputs and desired output. Assume that the desired output for a valid code is a 1, and for an invalid code is 0. Complete the truth table as shown in Table 1. A is the most significant bit, and D is the least significant bit.

b) Draw the Karnaugh map, and write the simplified Boolean expression for the valid codes as sum of products.

A	B	C	D	X
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

c) Draw the circuit for the above simplified Boolean expression.

d) Using the universal property of the NAND gate connect an equivalent circuit for these codes that uses only NAND gates.

Part 2: Boolean Functions (1)

1. Simplify the following two Boolean functions by means of Karnaugh maps.

$$F_1 (A, B, C, D) = \Sigma m (0,1,4,5,8,9,10,12,13)$$

$$F_2 (A, B, C, D) = \Sigma m (3,5,7,8,10,11,13,15)$$

2. Draw the logic diagrams for outputs F_1 and F_2 in terms of the inputs A, B, C, and D.

3. Implement and draw the two functions F_1 and F_2 together by using minimum number of NAND gates.

4. Connect the circuit and verify it's operation by preparing a truth table for F_1 and F_2 similar to Table 1.

Part 3: Boolean Functions (2)

1. Derive a truth table for the following Boolean Functions.

   $F=A'D+B'D+BC+AB'D$

2. Draw a Karnaugh map.

3. Combine all the 1's to obtain the simplified function for *F*.

4. Combine all the 0's to obtain the simplified function for *F'*.

5. Using logicWorks, implement both F and F' using NAND gates and connect two circuits to the same input switches but to separate output LED's. Prove that both circuits are complement of each other. In the lab implement and verify the operations of the circuit.

6. Draw both circuits.

Part 4: A Majority

A nine member legislative committee requires a 2/3 vote to spend a billion dollars. The vote is tabulated and converted to BCD code. If 2/3 of the committee is in favor, the vote will be the BCD representation of 6, 7, 8, or 9.

1. Derive a truth table for the problem, Table 2.

2. Derive a minimum sum of products expression from the map. {Enter the invalid BCD codes on the map as don't cares (x)}.

3. Using LogicWorks, design a circuit that lights an LED if a majority has voted in favor of spending the billion dollars. Implement this circuit and verify its operation in the lab using hardware.

EE 200 DIGITAL LOGIC CIRCUIT DESIGN EXPERIMENT #6

DESIGN OF CODE CONVERTERS

OBJECTIVE:

1. Design and build gray code to binary converter.

2. Design and build BCD-to-7 segment converter.

APPARATUS:

- Seven segment display.
- SN 7400 quad 2-input NAND gates (1)
- SN 7410 triple 3-input NAND gates (4)
- SN 7420 dual 4-input NAND gates (4)
- SN 7404 HEX inverter (1)
- SN 7446 BCD-to-seven segment decoder.

THEORY:

The conversion from one code to another is common in digital systems. Sometimes the output of a system is used as the input to the other system. A conversion circuit is necessary between 2 systems if each system uses different codes for the same information.

In this experiment you will design and construct 3-combinational circuit converters:

See section 4-5 in your book for further information.

Procedure:

1. ***Gray code to Binary converter:***

   Gray code is one of the codes used in digital systems. It has the advantage over binary numbers that only one bit in the code word changes when going from one number to the next. (See Table 1).

   Design a combinational circuit with 4 inputs and 4 outputs that converts a four-bit gray code number into an equivalent four-bit Binary number. Use Karnaugh map technique for simplification. Use LogicWorks for pre-lab demonstrations. Select the library "7400dev.clf" in the Parts Palette and then select the XOR chip 74-86. This would give you a set of 4 XOR's as shown in Fig. 1, just like the hardware chip 74-86. You could use as many as needed from these XOR gates in your design. Get back to ALL LIBRARIES and select switches for the inputs and Binary Probes as indicators of the outputs. Verify your design in the pre-Lab. During the Lab construct the circuit and verify its operations.

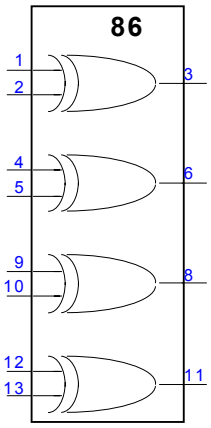


Figure. 1 XOR chip74-86

Table 1

Decimal	Gray	Binary
0	0000	0000
1	0001	0001
2	0011	0010
3	0010	0011
4	0110	0100
5	0111	0101
6	0101	0110
7	0100	0111
8	1100	1000
9	1101	1001
10	1111	1010
11	1110	1011
12	1010	1100
13	1011	1101
14	1001	1110
15	1000	1111

2. *BCD-to-seven Segment converter:*

A light emitting Diode (LED) is a PN junction diode. When the diode is forward biased, a current flows through the junction and the light is emitted. See Fig.2.

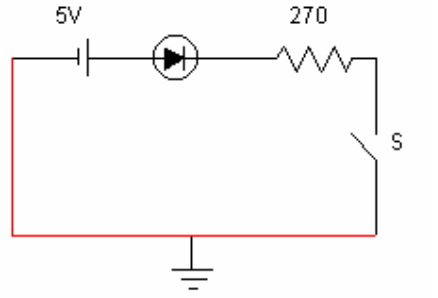


Figure.2

A seven segment LED display contains 7 LEDs. Each LED is called a segment and they are identified as (a, b, c, d, e, f, g) segments. Figure 3.

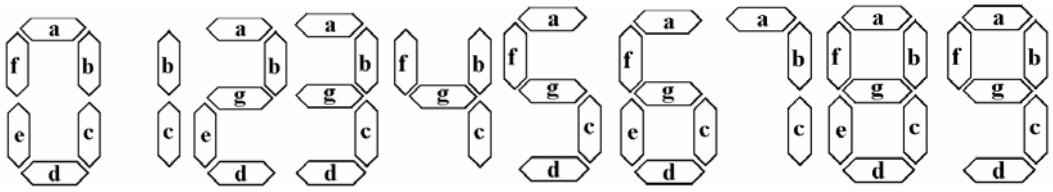


Figure 3. Digits represented by the 7 segments

The display has 7 inputs each connected to an LED segment. All anodes of LEDs are tied together and joined to 5 volts (this type is called common anode type). A limiting resistance network must be used at the inputs to protect the 7-segment from overloading.

BCD inputs are converted into 7 segment inputs (a, b, c, d, e, f, g) by using a decoder, as shown in Fig.4.

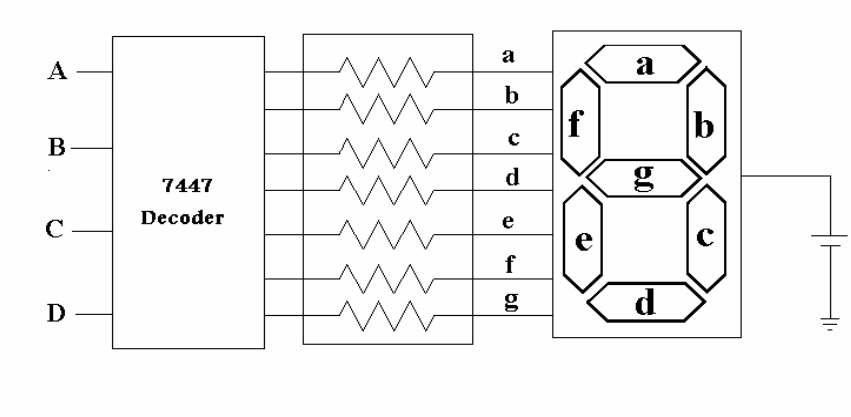


Figure. 4

A decoder is a combinational circuit that converts binary information from n input lines to a maximum of 2^n output lines. The input to the decoder is a BCD code and the outputs of the systems are the seven segments a, b, c, d, e, f, and g. For further information and pin connections, consult the specification sheet for decoder and 7-segment units.

First design a combinational circuit which would simulate the decoder function for only the segment "a", of the display. This can be done in the following steps:

a) Write down the truth table with 4 inputs and 7 outputs (Table 2)

b) For only the output "a", obtain a minimum logic function. Realize this function using NAND gates and inverters only. For example if decimal 9 is to be displayed a, b, c, d, f, g must be 0 and the others must be 1 (For common anode type display units), if decimal 5 is to be displayed then a, f, g, c, d must be 0 and the others must be 1.

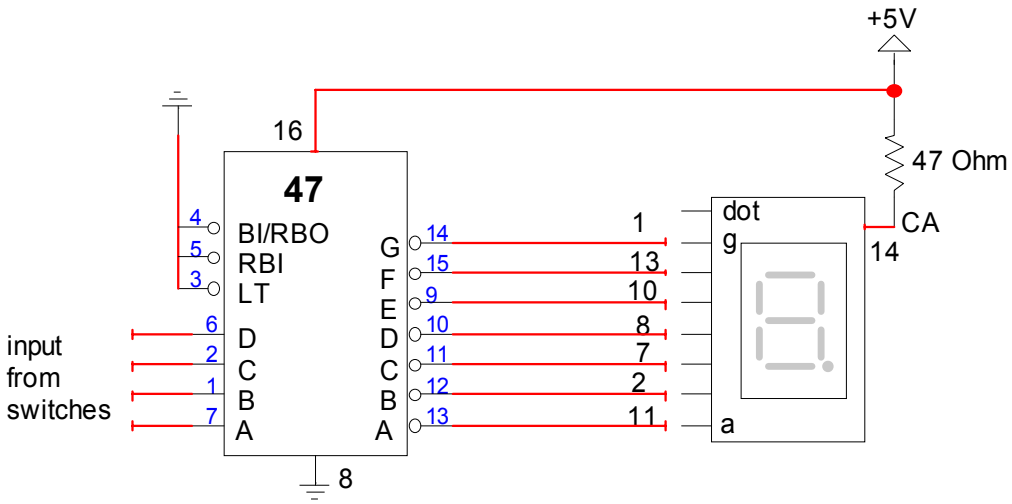
c) Connect the output "a" of your circuit to appropriate input of 7-segment display unit. By applying BCD codes verify the displayed decimal digits for that segment for "a" of the display.

d) Replace your circuit by a decoder IC 7447 for all of the seven segments. Observe the display and record the segments that will light up for invalid inputs sequence.

e) Comment on the design if you don't want to see any digit for invalid input sequence.

Table 2

Dec.	BCD				Outputs						
	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0							
1	0	0	0	1							
2	0	0	0	0							
3	0	0	1	1							
4	0	1	0	0							
5	0	1	0	1							
6	0	1	1	0							
7	0	1	1	1							
8	1	0	0	0							
9	1	0	0	1							



BCD-to-Seven Segment Decoder and 7-segment display

Note: In an actual 7-segment display the dot is on the left

EE 200 DIGITAL LOGIC DESIGN EXPERIMENT #7

ADDERS, SUBTRACTORS AND MAGNITUDE COMPARATORS

Objectives:

- To construct and test various adders and subtractor circuits.
- To construct and test a magnitude comparator circuit.

Apparatus:

- IC type 7486 quad 2-input XOR gates
- IC type 7408 quad 2-input AND gates
- IC type 7404 HEX inverter
- IC type 7483 4-bit binary adder
- IC type 7485 4-bit magnitude comparator.

Theory:

See Sections 1-5,4-3,5-2,5-4 of your textbook.

a) *Addition:*

IC type 7483 is a 4-bit binary adder with fast carry. The pin assignment is shown in Fig 1. The two 4-bit input binary numbers are A_1 through A_4 and B_1 through B_4. The 4-bit sum is obtained from S_1 through S_4. C_i is the input carry and C_o the out carry. This IC can be used as an adder-subtractor as a magnitude comparator.

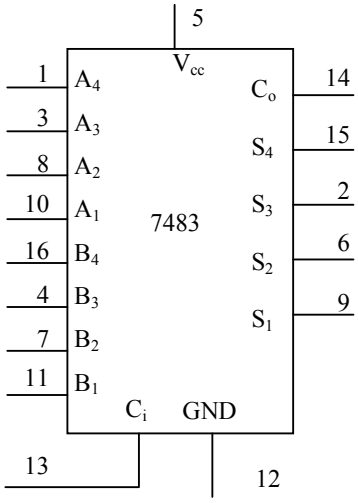
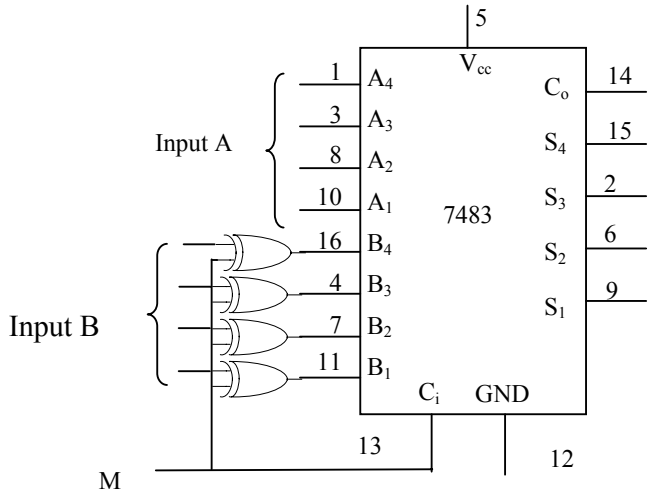


Fig.1 IC type 7483 4-bit adder

b) *Subtraction:*

The subtraction of two binary numbers can be done by taking the 2's complement of the subtrahend and adding it to the minued. The 2's complement can be obtained by taking the 1's complement and adding 1.

To perform A - B, we complement the four bits of B, add them to the four bits of A, and add 1 to the input carry. This is done as shown in Fig 2.



M = 0 for add and M = 1 for subtract

Fig. 2 4-bit adder/subtractor

Four XOR gates complement the bits of B when the mode select M = 1 (because $x \oplus 1 = x'$) and leave the bits of B unchanged when M = 0 (because $x \oplus 0 = x$) thus, when the mode select M is equal to 1, the input carry C_i is equal to 1 and the sum output is A plus the 2's complement of B. When M is equal to 0, the input carry is equal to 0 and the sum generates A + B.

c) *Magnitude comparison*

The comparison of two numbers is an operation that determines whether one number is greater than, equal to, or less than the other number.

The IC 7485 is a 4 bit magnitude comparator. It compares two 4-Bit binary numbers (labeled as A&B) generates an output of 1 at one of three outputs labeled A > B, A < B, A = B. Three inputs are available for cascading comparators. see Fig.3.

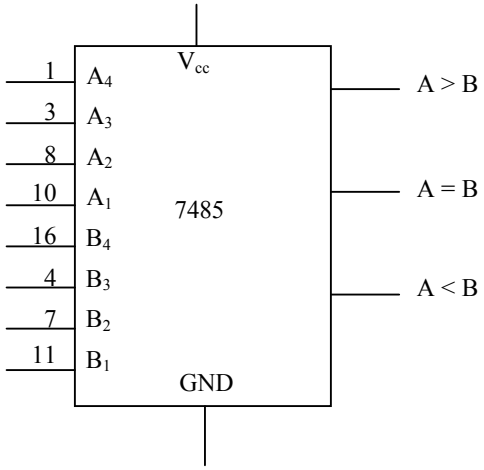


Fig. 3 4-bit magnitude comparator

Procedure:

a) Design using LogicWorks a half adder circuit using only XOR gates and NAND gates. Then during the Lab construct the circuit and verify its operation.
b) Design using LogicWorks a full adder circuit using only XOR gates and NAND gates. Then during the Lab construct the circuit and verify its operation.
c) Use IC 7483 to add the two 4-bit numbers A and B shown in Table1. In LogicWorks, select the chip 74-83 and use Binary switches for the bits of the two numbers and the input carry and use Binary Probe for the sum and carry out.

Table 1.

A3	A2	A1	A0	B3	B2	B1	B0	Sum				Carry out
1	0	0	1	0	0	1	0					
0	1	1	0	1	0	1	1					
1	1	0	0	1	0	1	0					

Input carry Ci is taken as logic 0. Show that if the input carry is 1, it adds 1 to the output sum.

In the Lab use switches S1-1 to S1-8 for the two numbers and use the SPDT S2 for the input carry Ci. For sum and carry out, use LED-1 to LED-5.

d) Connect the adder-subtractor circuit as shown in Fig 2. Perform the following operations and record the values of the output sum and the output carry C_o.

Table 2.

Decimal A B	**Output sum**				**Carry Out C_o**
9 + 5					
9 - 5					
9 + 13					
9 - 9					
10 + 6					
6 - 10					

- Show that C_o =1 when sum exceeds 15.
- Comment on sum and C_o for the subtraction operations when A > B and A < B.

e) Use IC7485 to compare the following two 4 bit numbers A and B. Record the outputs in table 3. Note that in LogicWorks you need to connect (A = B) input to logic 1 (as an indication that previous stages are equal in multi-digit numbers) for correct results while this is not necessary for the hardware.

Table 3.

A	**B**	**Outputs**
1001	0110	
1100	1110	
0011	0101	
0101	0101	

f) A magnitude comparator can be constructed by using a subtractor as in Fig 2. and an additional combinational circuit. This is done with a combinational circuit which has 5 inputs S_1, S_2, S_3, S_4, and C_o, and three outputs X, Y, Z see Fig.4

X = 1 if A = B Where S = 0000

Y = 1 if A < B Where C_o = 0

Z = 1 if A > B Where C_o = 1 S ≠ 0000

Design and construct this logic circuit with minimum number of gates. Check the comparator action using Part (e). In the Lab verify your LogicWorks simulation.

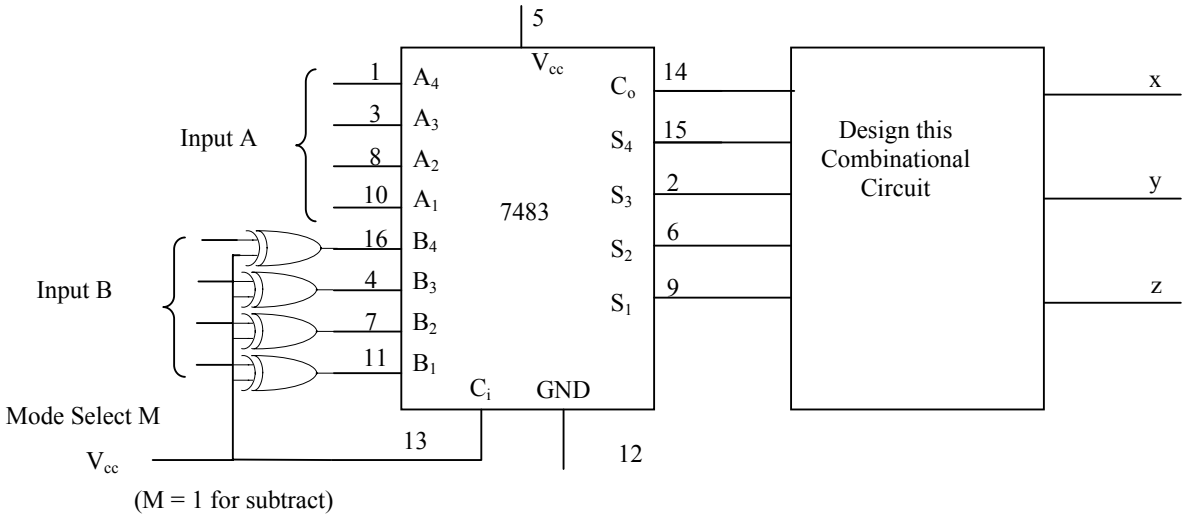


Fig.4 A magnitude comparator using a subtractor

EE 200 DIGITAL LOGIC DESIGN EXPERIMENT #8

DESIGN WITH MULTIPLEXERS

Objectives:

To design a combinational circuit and implement it with multiplexers. To use a demultiplexer to implement a multiple output combinational circuit from the same input variables.

Apparatus:

- IC type 7404 HEX inverter
- IC type 7408 quad 2-input AND gate
- IC type 74151 8x1 multiplexer (1)
- IC type 74153 dual 4x1 multiplexer (2)
- IC type 7446 BCD-to-Seven-Segment decoder (1)
- Resistance network (1)
- Seven-Segment Display (1)

Theory: see section 5.6 of your text.

IC Description:

74151 is a 8 line-to-1 line multiplexer. It has the schematic representation shown in Fig 1. Selection lines S_2, S_1 and S_0 select the particular input to be multiplexed and applied to the output.

Strobe S acts as an enable signal. If strobe =1, the chip 74151 is disabled and output y = 0. If strobe = 0 then the chip 74151 is enabled and functions as a multiplexer. Table 1 shows the multiplex function of 74151 in terms of select lines.

Table 1.

Strobe	Select Lines			Output
S	S_2	S_1	S_0	Y
1	X	X	X	0
0	0	0	0	D0
0	0	0	1	D1
0	0	1	0	D2
0	0	1	1	D3
0	1	0	0	D4
0	1	0	1	D5
0	1	1	0	D6
0	1	1	1	D7

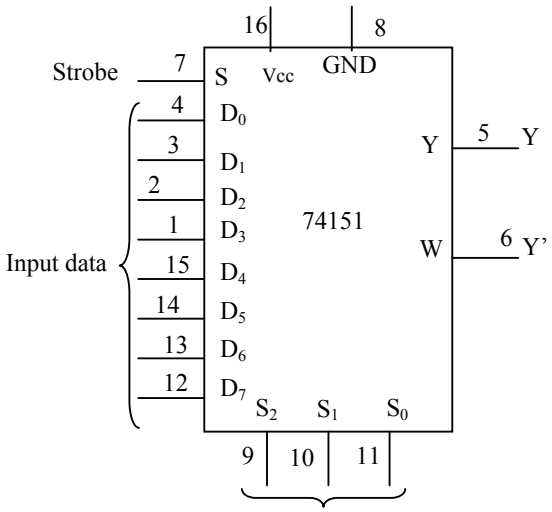


Fig.1 IC type 74151 Multiplexer 8×1

74153 is a dual 4 line-to-1 line multiplexer. It has the schematic representation shown in Fig 2. Selection lines S1 and S0 select the particular input to be multiplexed and applied to the output IY{1 = 1, 2}.

Each of the strobe signals $I\overline{G}$ {I = 1, 2} acts as an enable signal for the corresponding multiplexer.

Table 2. shows the multiplex function of 74153 in terms of select lines. Note that each of the on-chip multiplexers act independently from the other, while sharing the same select lines S_1 and S_0.

Table 2

	Multiplexer 1		
Strobe	Select lines		Output
1G	S_1	S_0	1Y
1	X	X	0
0	0	0	$1D_0$
0	0	1	$1D_1$
0	1	0	$1D_2$
0	1	1	$1D_3$

	Multiplexer 2		
Strobe	Select lines		Output
2G	S_1	S_0	2Y
1	X	X	0
0	0	0	$2D_0$
0	0	1	$2D_1$
0	1	0	$2D_2$
0	1	1	$2D_3$

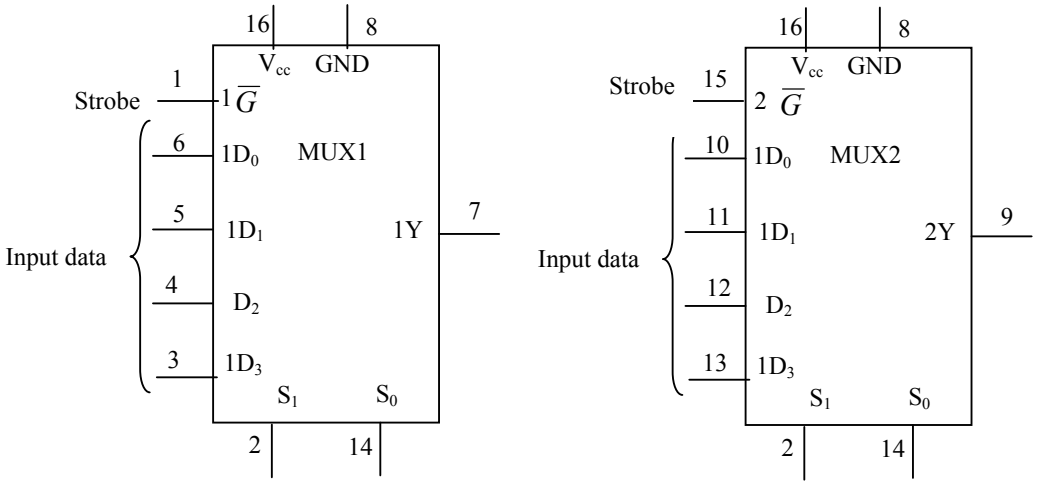


Fig.2 Chip 74153

IC 7446 is a BCD to seven segment decoder driver. It is used to convert the combinational circuit outputs in BCD forms into 7 segment digits for the 7 segment LED display units. See experiment #5.

Procedure:

Part I: Parity Generator:

a) Design a parity generator by using a 74151 multiplexer. Parity is an extra bit attached to a code to check that the code has been received correctly.

Odd parity bit means that the number of 1's in the code including the parity bit is an odd number. Fill the output column of the truth table in Table 2 for a 5-bit code in which four of the bits (A,B,C,D) represents the information to be sent and fifth bit (x), represents the parity bit. The required parity is an odd parity.

The inputs B,C and D correspond to the select inputs of 74151. Complete the truth table in Table 3 by filling in the last column with 0,1,A or A'.

b) Simulate the circuit using LogicWorks, use 74-151 multiplexer and Binary switches for inputs and Binary Probes for outputs. The 74151 has one output for Y and another inverted output W. Use A and A' for providing values for inputs 0-7. The internal values "A, B, C" are used for selection inputs B,C, and D. Simulate the circuit and test each input combination filling in the table shown below. In the Lab connect the circuit and verify the operations. Connect an LED to the multiplexer output so that it represents the parity bit which lights any time when the four bits input have even parity.

Inputs				Outputs	Connect data to
A	B	C	D	X	
0	0	0	0		
0	0	0	1		
0	0	1	0		
0	0	1	1		
0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

Part 2: Vote Counter:

A committee is composed of a chairman (C), a senior member (S), and a member (M). The rules of the committee state that:

- The vote of the member (M) will be counted as 2 votes
- The vote of the senior member will be counted as 3 votes.
- The vote of the chairman will be counted as 5 votes.

Each of these persons has a switch to close (“l”) when voting yes and to open (“0”) when voting no.

It is necessary to design a circuit that displays the total number of votes for each issue. Use a seven segment display and a decoder to display the required number.

If all members vote no for an issue the display should be blank. (Recall from Experiment #5, that a binary input 15 into the 7446 blanks all seven segments).

If all members vote yes for an issue, the display should be 0. Otherwise the display shows a decimal number equal to the number of 'yes' votes. Use two 74153 units, which include four multiplexers to design the combinational circuit that converts the inputs from the members’ switch to the BCD digit for the 7446.

In LogicWorks use +5V for Logic 1 and ground for Logic 0 and use switches for C, S, and M. Use two chips 74153 and one decoder 7446 verify your design and get a copy of your circuit with the pin numbers to Lab so that you could connect the hardware in exactly the same way.

DESIGN WITH ROM'S

(Handout will be provided)

EE 200 DIGITAL LOGIC DESIGN EXPERIMENT #10

FLIP-FLOPS

Objectives:

1. To become familiar with flip-flops.

2. To implement and observe the operation of different flip-flops.

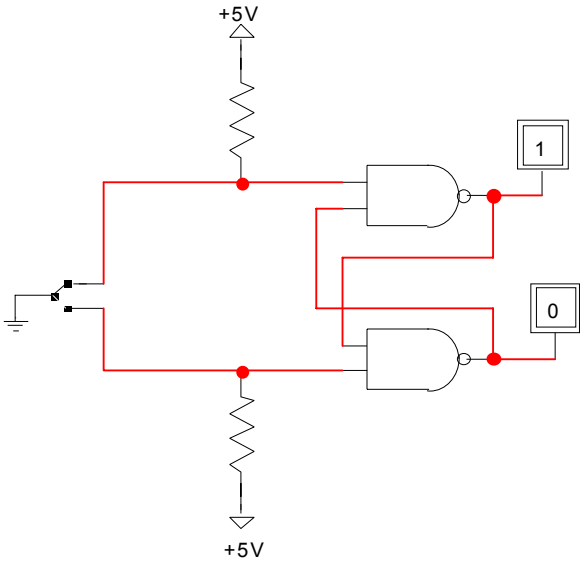
Apparatus:

- IC type 7400 quad 2-input NAND gate
- IC type 7410 triple 3- input NAND gate
- IC type 7476 dual JK master-slave flip-flops.
- IC type 7474 dual D positive-edge-trigged flip-flops.
- Dual trace oscilloscope.

Theory: See sections 6-2 and 6-3 of your text.

Procedure:

1. In the pre-lab using LogicWorks construct the circuit shown in Fig.1



Where we could use generic NAND gates or 74-00 and Binary Probes to simulate LEDs. Finally, we use SPDT for the bouncing switch. Using the simulated circuit fill in the truth table.

S	R	Q	Q’
1	0		
1	1		
0	1		
1	1		
0	0		

In the Lab, Build the RS latch shown in fig.2. Use SPDT switch S2 as a bouncing switch. Q and Q’ Outputs are connected to LED’S of the PB-503. Verify the truth table experimentally.

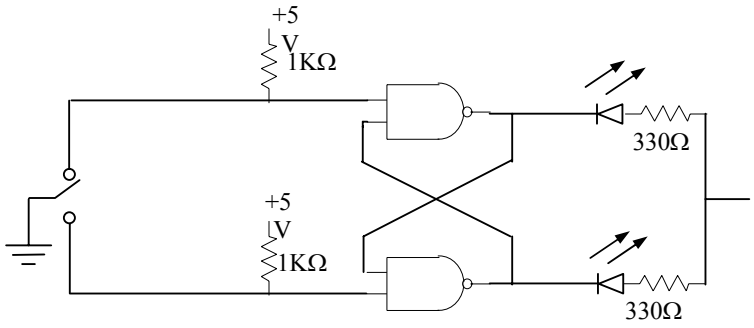


Fig. 2

2. Modify the basic R-S into a D latch by adding the steering gates and the inverter shown in Fig 3.

   Connect the D input to the pulse generator of the digi designer and set it at 1 Hz.

   Connect the enable input to a high through 1k resistor. Observe the output; obtain the truth table experimentally then change the enable to a low.

Is the enable an active high or an active? Leave the enable low and place a momentary short to ground first on one output and then on the other. What happens?

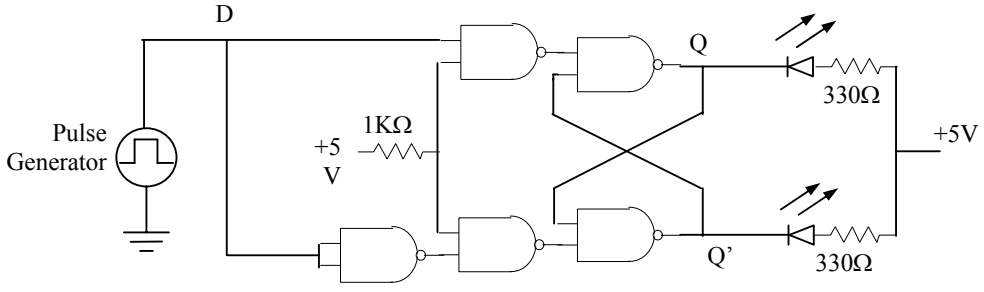


Fig. 3

3. The 7476 is a dual JK master-slave flip-flops with preset and clear inputs. The function table given in table 1 defines the operation of the flip-flop. The +ve transition of the CLOCK (CP) pulse changes the master flip-flop, and the (-ve)

transition changes the slave flip-flop as well as the output of the circuit. In LogicWorks the chip 7476 is not available, however, the generic JK flip-flop behave in exactly the same way as the 7476. The "S" represents the Preset, the "R" represents the Clear, and C represents the clock pulse (CP). Verify the table by connecting Binary switches to R, S, J, K, and C. Notice that only the negative edge of the clock affects the outputs (Q, and Q').

Table 1

Input					Output	
Preset	Clear	Clock	J	K	Q	Q'
0	1	X	X	X	1	0
1	0	X	X	X	0	1
0	0	X	X	X	1	1
1	1	↓	0	0	No change	
1	1	↓	0	1	0	1
1	1	↓	1	0	1	0
1	1	↓	1	1	Toggle	

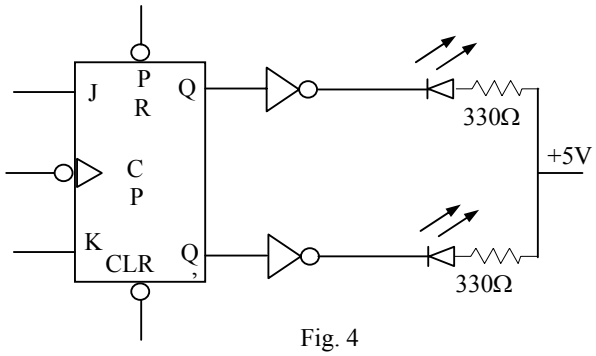


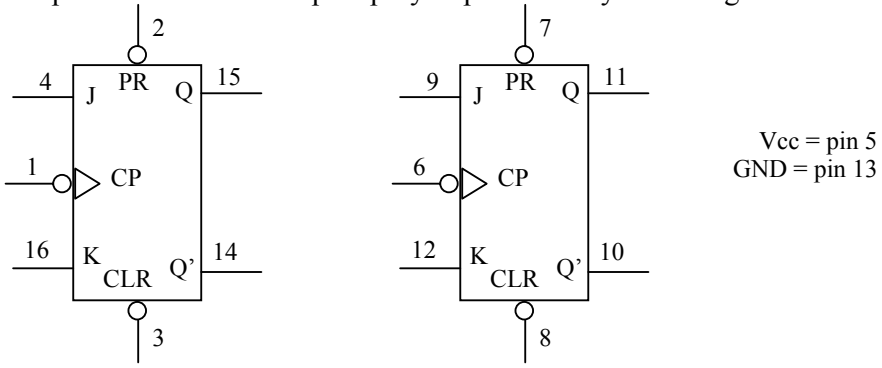
Fig. 4

In the Lab, Construct the circuit of Fig 4. Look at the data sheet for the 7476 and determine the inactive logic required at the PRE and CLR inputs.

Connect the 7476 for the SET mode by connecting J = 1, K = 0. With CLOCK (CP) = 0; test the effect of PRE, CLR by putting a 0 on each, one at a time.

Put CLR = 0, then pulse the clock (CP) by putting a HIGH then a LOW, on the clock. Does the CLR input override J input?

Verify the operation of the JK flip flop by experimentally obtaining the characteristic table.



CLOCKED SEQUENTIAL CIRCUITS AND COUNTERS

OBJECTIVE:

- To design, build and test synchronous sequential circuits.
- To design, build, and test synchronous counters
- To design, build and test asynchronous counters

APPARATUS:

- IC type 7476 dual JK master-slave flip-flops
- IC type 7400 quad 2-input NAND gates

THEORY:

See sections 6-6, 6-7, 6-8, 7.2 and 7.5 of your own text.

PROCEDURE:

1. *SYNCHRONOUS SEQUENTIAL CIRCUITS:*

a) Design, construct and test a sequential circuit whose state is shown in Fig.1. Use JK flip-flops in the design.

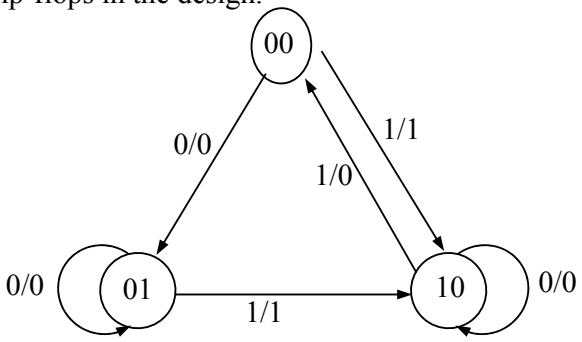


Fig. 1

The circuit has two flip-flops A, B, one input x and one output y. The circuit is to be designed by treating the unused states as don't care conditions. The final circuit must be analysed to ensure that it is self-correcting. If not suggest a solution.

b) Complete the excitation table shown in Table 1.

Table 1.

Present state		Input	Next state		Output	Flip-flop input functions			
A	B	X	A	B	Y	JA	KA	JB	KB
0	0	0	0	1	0	0	X	1	X
0	0	1	1	0	1	1	X	0	X
0	1	0							
0	1	1							
1	0	0							
1	0	1							
1	1	0							
1	1	1							

c) Using Karnaugh maps obtain minimal expressions for the flip-flop input functions JA, …, KC

d) Simulate the circuit using LogicWorks. LogicWorks does not have the JK master-slave flip-flop IC 7476. Use instead the generic JK flip-flop as you did in experiment 9. In the Lab, build the circuit and check the output to verify the state table values.

2. *Synchronous Counters*

Synchronous counters have all clock lines tied to a common clock causing all flip-flops to change at the same time. The count sequence of a counter can be analysed by placing the counter into every possible number in the sequence and determining the next number in the sequence state diagram is developed as the analysis proceeds. (A state diagram is an illustration of the transitions that occur after each clock pulse).

a) In the pre-lab using LogicWorks and then in the lab using hardware chips, design a 2-bit gray code counter using JK flip-flops. The required sequence is the binary equivalent of (0-1-3-2-0). A state diagram for this counter is given in Fig. 2.

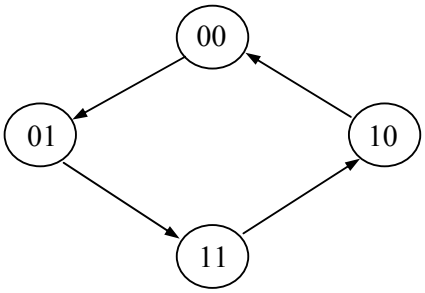


Fig. 2

b) Complete the excitation table (Table 2) for the counter and obtain logic expression for the JK flip-flop input functions.

Table 2.

Present state		Next state		Flip-flop input functions			
A	B	A	B	JA	KA	JB	KB
0	0						
0	1						
1	1						
1	0						

Flip-flop input functions are:

JA= KA=

JB= KB=

c) In the lab, build the circuit and test it by pulsing it from the PB-503. Check that the output is the designed sequence

3. *A Synchronous Counters*

Asynchronous counters are a series of flip-flops each clocked by the previous state, one after the other. Since all the stages of the counter are not clocked together, a ripple effect propagates as various flip-flops are clocked. For this reason they are called ripple counters. The modulus of a counter is the number of different output states the counter may take (i.e. Mod 4 means the counter has four output states).

a) In the pre-lab construct a 4-bit asynchronous counter shown in Fig.3. (It is also called binary ripple counter). Use four generic JK flip-flops. Connect four Binary Probes to Q outputs. Connect all R and S inputs to Logic 1 and connect a switch to the CP input.

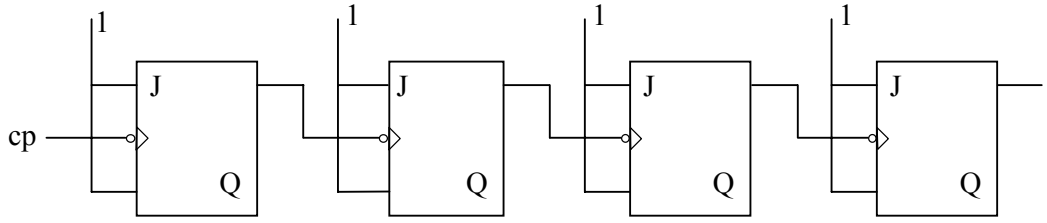


Fig. 3 4-bit ripple counter

b) In the Lab use two 7476 ICs to implement the design. Connect Q outputs of flip-flops to indicator lamps of the PB-503. Connect all clear (CLR) and preset (PRE) inputs to logic 1. Connect the CP input to the pulse output of the PB-503 and check the counter for proper operation.

c) Write down the count sequence in Table 3. Identify this count sequence (up or down). Comment on what happens after the application of 15 pulses to CP input.

Table 3. Count sequence for the 4-bit ripple counter.

A	B	C	D

Appendix

(LABORATORY REGULATIONS AND SAFETY RULES)

The following Regulations and Safety Rules must be observed in all concerned laboratory location.

1. **It is the duty of all concerned who use any electrical laboratory to take all reasonable steps to safeguard the HEALTH and SAFETY of themselves and all other users and visitors.**
2. **Be sure that all equipment is properly working before using them for laboratory exercises. Any defective equipment must be reported immediately to the Lab. Instructors or Lab. Technical Staff.**
3. **Students are allowed to use only the equipment provided in the experiment manual or equipment used for senior project laboratory.**
4. **Power supply terminals connected to any circuit are only energized with the presence of the Instructor or Lab. Staff.**
5. **Students should keep a safe distance from the circuit breakers, electric circuits or any moving parts during the experiment.**
6. **Avoid any part of your body to be connected to the energized circuit and ground.**
7. **Switch off the equipment and disconnect the power supplies from the circuit before leaving the laboratory.**
8. **Observe cleanliness and proper laboratory house keeping of the equipment and other related accessories.**
9. **Wear proper clothes and safety gloves or goggles required in working areas that involves fabrications of printed circuit boards, chemicals process control system, antenna communication equipment and laser facility laboratories.**
10. **Double check your circuit connections specifically in handling electrical power machines, AC motors and generators before switching "ON" the power supply.**
11. **Make sure that the last connection to be made in your circuit is the power supply and first thing to be disconnected is also the power supply.**
12. **Equipment should not be removed, transferred to any location without permission from the laboratory staff.**
13. **Software installation in any computer laboratory is not allowed without the permission from the Laboratory Staff.**
14. **Computer games are strictly prohibited in the computer laboratory.**
15. **Students are not allowed to use any equipment without proper orientation and actual hands on equipment operation.**
16. **Smoking and drinking in the laboratory are not permitted.**

All these rules and regulations are necessary precaution in Electrical Laboratory to safeguard the students, laboratory staff, the equipment and other laboratory users.